#3

Europäisches Patentamt	European Patent Office	Office européen des brevets

Bescheinigung | Certificate | Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. | Patent application No. | Demande de brevet n°

00100505.7

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE, 01/02/01
LA HAYE, LE

EPA/EPO/OEB Form 1014 - 02.91

BEST AVAILABLE COPY

Europäisches Patentamt
European Patent Office
Office européen des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.:
Demande n°: 00100505.7

Anmeldetag:
Date of filing:
Date de dépôt: 11/01/00

Anmelder:
Applicant(s):
Demandeur(s):
Hewlett-Packard Company
Palo Alto, California 94304
UNITED STATES OF AMERICA

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
System and method for service and support automation of a network interconnecting environment

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: State: Pays:	Tag: Date: Date:	Aktenzeichen: File no. Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:
/

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE/TR
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

Hewlett-Packard Company

January 11, 2000

Our Ref.: 20-99-0041

SYSTEM AND METHOD FOR SERVICE AND SUPPORT AUTOMATION OF A NETWORK INTERCONNECTING ENVIRONMENT

BACKGROUND OF THE INVENTION

The present invention relates to a system for providing network support services, particularly for electronically providing network configuration services, between a network service provider and a proprietary network comprising a network interconnecting infrastructure. Further, the invention relates to a method for providing such network support services, particularly for electronically providing network configuration services, between a network service provider and a proprietary network comprising a network interconnecting infrastructure, whereby the network service provider's site and the proprietary network's site being spatially separated from each other.

For most companies the network (Internet or Intranet) is the most critical and often the most complicated element of their entire IT infrastructure, yet for many organisations knowing how their network is performing and planning expansion to meet the changing business requirements are often overlooked. Proprietary or customized networks therefore have to be maintained by way of support services in order to maximize investment. These support services are delivered by technical specialists, either locally or remotely.

For support services to a local area network (LAN), there exist network management systems which provide detailed LAN health checks utilising passive monitoring probes. Full analysis and any errors or capacity problems identified are documented in a report. They further provide full time remote LAN monitoring via network probes and ISDN connections. Monthly reports detail utilization errors, protocol use, and all SNMP (Simple Network Management Protocol) based products. A recommendation on solutions to any problems

identified is hereby included. In addition, they provide a remote or local diagnosis service enabling the customer to get experienced network specialists connected and diagnosing problems using network analysers. They accordingly provide remedial service on all hardware elements of the customers network which includes unit replacement where applicable, and reloading of configuration. The service includes initial problem diagnosis, alarm recording, performance analysis and reporting or problems to the customer. All these services can be handled remotely whereby a customer's network is managed from a network service provider's management center.

Web-based management capabilities are already known in enterprise management platforms, but have since migrated to LAN management platforms. The present applicant which offers Web interfaces to its enterprise-oriented OpenView ™ Network Node Manager, has extended that capability to a LAN-oriented Professional Suite. In particular, a web browser is shipped together with the OpenView Professional Suite, which is a comprehensive software solution that allows customers in small to midsize networked environments to manage virtually all elements of a PC LAN.

The enhanced version of the OpenView ™ Professional Suite can automatically discover Web devices. It can also provide a menu that lets the administrator launch a Web browser to display the management page of any device that can be managed via Web technology. With the Web interface, a network administrator is able to access Web-based LAN management information from any location, such as a remote server site or a home computer.

Thus the OpenView ™ Professional Suite combines the power of a central management console, including automatic discovery and map generation, with the ease of using the Web to manage devices. OpenView provides direct access from an OpenView map to the management page of devices that can be managed through any standard Web browser. It accomplishes this by

automatically discovering these devices and providing menu items that launch a Web browser with the appropriate page displayed.

Thereupon, known network interconnect support and management systems for reliability control, in particular those systems for managing network interconnect devices, like routers, therefore can be used to assist in the following areas:

- Reliability improvement
- Performance enhancement
- Capacity management
- Fault diagnosis & rectification

These systems identify & diagnose problems before they impact the business or handle routine tasks in order to maintain the network.

In contrast to the above approaches, there do not exist any pertinent network support systems comprising the prementioned features but which can be used for service and support automation in an Enterprise-wide network environment or infrastructure. On the other hand, particularly in case of these network environments, service and support automation is essential too to handle network availability and network complexity at lower cost. Thereupon, network interconnect device performance optimization is required in order to maximize network throughput and to correspondingly minimize network maintenance costs and network idle times (down-time).

SUMMARY OF THE INVENTION

It is an object of the invention to provide a system and a method for network service and support services which allows for an effective service and support automation particularly with the highest degree of automatisation as possible, not at least in order to minimize the necessary staff of service engineers.

Another object is to provide such a system and method which allow for simplified remote management of a network interconnecting infrastructure i.e. for a flexible customized service which can be delivered on or off site.

A further object is to provide such a system and method which maximises network availability of a serviced and supported network environment, particularly a network interconnecting environment.

It is still another object to provide such a system and method which allow for mission critical end to end and secure utility management of such a network or a network interconnecting structure, respectively.

The object is solved by the independent claims 1 and 10. Preferred embodiments are shown by the dependent claims. Particularly there are provided information collecting means, located at the proprietary network's site, for collecting configuration information about the network interconnecting infrastructure which allows for an automatic capture of configuration information via a data collection mechanism which is independent of HW devices. The proposed mechanism can advantageously be running as a background task. Further provided are information transferring means, located at least at the proprietary network's site, for transferring the collected configuration information from the proprietary network to the network service provider. In addition, database means are provided which are located at the network service provider's site, for storing a data model modeling at least part of the entire network interconnecting infrastructure of the proprietary network. Preferably, the data model is modeling a whole network interconnecting

infrastructure including but not limited to network interconnect hardware and related software drivers. Thereupon, new high edn services can be provided through integration of hardware, operating system, network and application data into only one data model. Based on that data model, information processing means, located at the network service provider's site, consolidate the collected configuration information in the data model, analyze the consolidated data model, and generate a network support service based on the results of the analyzing of the consolidated data model.

In a preferred embodiment, the information transferring means comprise transport managing means whereby the collected configuration information is transferred via an information network, particularly the Internet, or by means of a data carrier. Network support service therefore can be handled as an electronic service as part of electronic commerce and business. The proposed web-based approach guarantees compatibility, platform-independency and high accessability.

In another embodiment, the collected configuration information is being stored in a data file, in particular a plain ASCII file, which is then transferred from the proprietary network's site to the network service provider's site. Storing configuration information in a plain data file particulary provides an (open) architecure or technology which is open or ready for new collection protocol releases as new releases can easily be inserted in the ASCII file. Further, a plain ASCII file is advantageously operating system or hardware independent. Thereupon, backup and restore of configuration information enables fast diagnosis and repair and thus high network availability.

In another embodiment, scalable database means, in particular an object-oriented and/or relational database, and a corresponding scalable data model are provided. Using a scalable database gurantees high scalability of the stored data model in view of a growing network enviroment. A relational database

particularly provides for hardware environment consistent growing of the data model.

According to still another embodiment, the database means include means for storing performance history information for the network interconnecting infrastructure. This allows for monitoring and/or analyzing a network interconnecting infrastructure and to find out whether the network performance can be enhanced through updates of the interconnecting hardware and related software drivers. Further, history information enables better diagnosis and performance check. Hereby, according to a further embodiment, the information processing means monitor and/or analyze the configuration, or configuration changes, and/or performance ot the network interconnecting infrastructure of the proprietary network, particularly based on rules. Additionally enabled is network health monitoring including but not limited to trend analysis, forecasts, traffic assessment and problem prediction. Therefore automatic monitoring of configuration changes with generic access to all kinds of data elements from all kinds of devices is enabled. Further, the configuration change monitoring enables detection of configuration differences which information can advantageously be used for network interconnecting failure detection. In addition, the proposed system architecture and process allow for an automated knowledge capture for configuration checking, trending and an enhanced failure prediction.

In order to optimize data transfer of collected information, not at least in view of the well-known data transfer bottlenecks of public networks like the Internet, and/or in order to avoid any impacts on the performance of a network caused by an active data collection process, means for scheduling the collection of the configuration information are provided. Hereby means for determining, particularly in advance, the configuration information which is to be collected can be advantageously provided. The effectiveness and performance of the data collection process can be considerably enhanced by providing a data

collection strategy.

Beyond this, the proposed system and method provide a remote electronic service (e-Service) over the Internet using electronic mail or the Web. Since the configuration data is stored in only one file, it can advantageously be transferred by way of electronic mail (e-mail) or via the Internet.

BRIEF DESCRIPTION OF THE DRAWINGS

Other objects and many of the attendant advantages of the present invention will be readily appreciated and become better understood by reference to the following detailed description when considering in connection with the accompanied drawings. Features that are substantially or functionally equal or similar will be referred to with the same reference sign(s).

Fig. 1 shows an architectural overview of an exemplary embodiment of a system according to the invention;

Fig. 2 shows a schematic overview of a data collector and its architectural environment according to a preferred embodiment;

Fig. 3 shows a more detailed functional architecture of an data collector according to the invention depicting three different layers;

Fig. 4 shows a functional architecture of a collection configuration according to the invention;

Fig. 5 shows a functional architecture of a transport office manager in accordance with an embodiment of the invention;

Fig. 6 shows a flowchart illustrating procedural steps during insertion of a new task;

Fig. 7a, b show an exemplary scheduling (a) and applying strategy (b) according to the invention by means of a task execution process flowchart;

Fig. 8 shows exemplary interfaces for collectible definitions;

Fig. 9a-c show exemplary strategy interfaces (a), exemplary interfaces for schedules (b) and interfaces for device information according to an embodiment of the invention; and

Fig. 10 shows an exemplary DMI collector and corresponding algorithm in detail.

DETAILED DESCRIPTION OF THE INVENTION

Fig. 1 shows an architectural overview of an exemplary embodiment of a system according to the invention. The architecture comprises a 'remote connectivity network documentation tool' on a customer site and a 'network support repository tool' on a service provider site which are linked e.g. via an ISDN link.

The architecure allows for automation of the necessary steps: get inventory, configuration and performance details from networks components, consolidate the gathered data in a data model, which is a combined generic data model for hardware, software and network; in addition, data condensing with statistic components can be accomplished in order to minimize the storage requirements for the transferred data sets (ASCII files).

The shown Web-based approach enables enhanced accessability and

particularly e-services and/or (e-)brokering. Preferably, configuration data is transferred via the Internet (i.e. e-Mail) or via post mail (e.g. by use of floppy disk). Further it allows for knowledge for consistency and health checks and allegedly apply repair actions or escalation to the client or a service engineer of the service provider.

Fig. 2 shows a preferred embodiment of the proposed data collector which depicts the principal architecture and functionality of a data collector and its architectural environment including its interfaces. The data collector collects configuration information of the network interconnecting environment which is data concerning network topology, nodes, hardware interconnecting devices (routers, switches) etc. and, beyond that, collecting configuration data for network components not exclusively including network hardware, software and/or topology info Infomation technology (IT) or computer network, telecommunicationnetwork, Internet, Intranet, or the like. It allows for the following functional options:

- Collection on demand (immediate and synchron), collection
- Collection according to a schedule on a regular (e.g. periodic) basis
- Support of several network management protocols like SNMP, DMI (Desktop Management Interface), WEBM (Web-Based Enterprise Management), etc.)

Particularly it can run as a background task. The provided interfaces which can be accessed by a user or network support manager of the data collector are

1. As part of the specification of collection tasks:

- The definition of what shall be collected
- Which device-device identification shall be used

- Specification of schedules per collectable

- How or where to deliver the result, either per notification or transmission, and

2. The initiation of a data collection procdure.

The proposed data collection mechanism includes configurable measurement modules using standard protocols.

Fig. 3 shows a more detailed functional architecture of an data collector according to the invention. The small circles depicted right-hand to the small boxes indicate COM components. The shown data collector splits into three layers. The first layer contains the Collection Protocols including but not limited to SNMP group information and tables, DMI information and predefined configuration files for specifically known interconnect devices. The second so-called "Strategy" layer defines the collection strategies which can be a sequential data collection, e.g. in a batch mode, or a parallel data collection, i.e. gathering several collectables simultaneously. The third layer is providing basic services of the exemplary data collector are a Collection Scheduler and cooresponding Schedule for the collectables and a TFTP server for interconnect configuration file collection.

The following is a preferred sequence of a collection task:

- ***What*** – Definition of Collectable

- ***Which Device*** – device Identification

- ***How often*** – Schedule

- ***Start*** – Task Initiation

- ***Forward*** – Start Scheduler with Task

- ***Schedule*** – Get Collection Time
- ***Plan*** – Strategy of Collection , defined by Collection Protocol
- ***Collect*** – Perform Collection with Collection Protocol
- ***Result*** – Deliver Result of the Collection to Application

The collection configuration for the collectables, e.g. systemgroup information, interface tables, configuration files, routing tables health parameters, etc., is related to the definition of device classes, the definition of the collectables and the relations between the device classes and the collectables.

Fig. 4 shows a functional architecture of a collection configuration according to the invention. The depicted boxes with chamfered edges indicate configuration file parsing done with TCL (Tool Control language). The following is a preferred sequence of a configuration task:

- Get a collectable for a given device
- Forward collection task to protocol specific collector

Fig. 5 shows a functional architecture of a transport office manager (TOM) in accordance with an embodiment of the invention. The following are preferred sequences of sending and receiving data.

<u>Sequence of Sending data:</u>

- Create a file (ASCII) for sending
- Notify TOM to send the file

<u>Sequence of Receiving data:</u>

- Notification about received files

- Retrieve file for further processing

In the following, the functionality of the collector components are described in more detail, in particular the control flow during the collection. The collection is a two-step process. In the first step, a client application configures a collection task and passes the task to the collector. The flowchart depicted in Fig. 6 shows what happens when a new collectible is inserted by the client, i.e. the procedural steps during insertion of a new task.

When the client application adds a new task to the collector, the collector checks whether the collection task is valid. In this embodiment, validity means that the task has configured the following attributes:

- A Schedule;
- A valid collectible definition;
- Device Information that contains the IP address and other access parameters for the collectible;
- A non null session identification that defines the application that defined the task.

If the task is not valid, it is rejected with an error message. The next step is to check whether the collectible definition matches the collector, e.g. that the SNMP collector gets only SNMP collectible definitions and not DMI collectibles. The task is rejected when the collectible does not match the collector. In the positive case the collector forwards the task to its scheduler. The scheduler dertermines the date and time of the first collectionand inserts the task into its queue.

The scheduler has an internal priority queue that holds a list of all collection tasks sorted by time. When a collection task is ready, the steps shown in the

flowchart depicted in Fig. 7a, which is the first part of an entire flow chart continued in Fig. 7b, are executed. It is noteworthy that the tasks are executed for every collection task that has to be performed. Fig. 7a, b particularly show an exemplary scheduling (a) and applying strategy (b) according to the invention by means of a task execution process flowchart.

At the beginning, the scheduler removes the task from the priority queue and determines the next collection time. Sometimes there will be no next collection time, e.g. in case of a collect once schedule. If the collection time exists it will insert the task with the new collection time again. Otherwise the task will not be inserted into the queue and therefore not handled again (i.e. collect once).

The next step checks whether the task should be forwarded to a corresponding strategy. If the task was suspended due to repeated errors, the scheduler will check whether to restart the task again. If it should be disbaled furthermore, the task is finished. Otherwise the scheduler will change the status of the task to 'active' and pass pass the task to the strategy. If the task was not suspended due to errors, it may be suspended then. Otherwise the task will be forwarded to the corresponding strategy.

The strategy holds a list of all collection tasks that have to be performed as fast as possible and passes the tasks, in accordance with the strategy, to the respective collection method. The collection method tries to retrieve the collectible. If the collection succeeds, a retry counter is reset which is used for suspending taks that resulted in errors for several times. Further it delivers the result and notifies it to the client if applicable. If the collection fails, the strategy increments the retry counter. The task will be suspended if the counter reaches a maximum.

The following Figures show the interfaces and their relationships to each other. In particular Fig. 8 shows exemplary interfaces for collectible definitions. Fig.

9a - c show exemplary strategy interfaces (a), exemplary interfaces for schedules (b) and interfaces for device information according to an embodiment of the invention.

Now referring to Fig. 9b, the first group of interfaces defines collection schedules. The collection schedules are used by the scheduler in order to find out when to perform a collection task. The interface basically provides two methods. One returns the date for the 'first' collection and the second method returns the date of the 'next' collection.

Referring now to Fig. 8, another group of collectible definition interfaces describes 'what' to collect. Common to all collectible definitions is a 'name' and a 'unique identifier'. Protocol specific information is provided by derived protocol specific interfaces. Fig. 8 gives an overview of the exemplary collectible definitions. For instance, the interfaces 'ISNMPCollectibleDef' define items that can be retrieved via SNMP.

Referring to Fig. 9c, a family of device information interfaces define how to access a device. The basic interface contains only the network address of the device. Protocol specific information like SNMP community strings, retry and timeouts are defined in protocol dependent interfaces. It is noted that the device information is protocol dependent. For example, an SNMP collection task needs the SNMP community strings. These strings are not provided by the base interface 'IDeviceInfo'. In order to define SNMP collection tasks, the interface 'ISNMPDeviceInfo' is to be used. This interface is derived from the interface 'IDeviceInfo' and extended by commonly known methods (algoritms) to retrieve and set the community strings.

Referring now to Fig. 9a, strategies are used to control access to a device. The base interface 'IcollectionStrategy' consists of two methods. The method 'CollectionMethod' sets the collection method that is used in conjunction with

that strategy. The collection method is preferably part of the protocol. The interface 'IparallelCollectionStrategy' is an inherited interface from 'IcollectionStrategy'. It has additional methods to set the maximum number of threads and to retrieve the number of currently active threads.

Finally, Fig. 10 shows an exemplary DMI collector and corresponding algorithm in greater detail. The DMI collector module retrieves arbitrary DMI groups and DMI tables. A DMI collectible is defined by the above described interface IDMICollectibleDef'. A DMI collectible is particularly defined by the following attributes:

- The component name;
- The class name for the DMI group or table;
- A list of IDs.

The DMI collector performs the following steps for each collectible:

- Enumerate all components fo a device
- For each component that matches the component name in the collectible definition:
    - Enumerate all DMI classes in the component;
    - For each class that matches the class name in the collectible definition:
        - Collect the item and return the result.

Fig. 10 shows the algorithm in detail.

Hewlett-Packard Company **January 11, 2000**
Our Ref.: 20-99-0041

CLAIMS

1. System for providing network support services, particularly for electronically providing network configuration services, between a network service provider and a proprietary network comprising a network interconnecting infrastructure,

characterized by

information collecting means, located at the proprietary network's site, for collecting configuration information about the network interconnecting infrastructure;

information transferring means, located at least at the proprietary network's site, for transferring the collected configuration information from the proprietary network to the network service provider;

database means, located at the network service provider's site, for storing a data model modeling at least part of the entire network interconnecting infrastructure of the proprietary network;

information processing means, located at the network service provider's site,

for consolidating the collected configuration information in the data model,

for analyzing the consolidated data model, and

for generating a network support service based on the results of the

analyzing of the consolidated data model.

2. System according to claim 1, characterized in that the information transferring means comprise transport managing means whereby the collected configuration information is transferred via an information network, particularly the Internet, or by means of a data carrier.

3. System according to claim 1 or 2, characterized in that the collected configuration information being stored in a data file, in particular a plain ASCII file, which is then transferred from the proprietary network's site to the network service provider's site.

4. System according to one or more of claims 1 to 3, characterized by scalable database means, in particular an object-oriented and/or relational database, and a corresponding scalable data model.

5. System according to one or more of the preceding claims, characterized in that the database means include means for storing performance history information for the network interconnecting infrastructure.

6. System according to one or more of the preceding claims, characterized in that the information processing means monitor and/or analyze the configuration, or configuration changes, and/or performance ot the network interconnecting infrastructure of the proprietary network, particularly based on rules.

7. System according to one or more of the preceding claims, characterized by means for scheduling the collection of the configuration information.

8. System according to one or more of the preceding claims, characterized

by means for determining, particularly in advance, the configuration information which is to be collected.

9. System according to claim 7 or 8, characterized in that providing a data collection strategy.

10. Method for providing network support services, particularly for electronically providing network configuration services, between a network service provider and a proprietary network comprising a network interconnecting infrastructure, whereby the network service provider's site and the proprietary network's site being spatially separated from each other,

characterized by the steps of

collecting configuration information about the network interconnecting infrastructure at the proprietary network's site,

transferring the collected configuration information from the proprietary network's site to the network service provider's site,

storing a data model modeling at least part of the entire network interconnecting infrastructure of the proprietary network in a database located at the network service provider's site,

consolidating the collected configuration information in the data model,

analyzing the consolidated data model, and

generating a network support service based on the results of the analyzing of the consolidated data model.

11. Method according to claim 10, characterized by transferring the collected configuration information via an information network, particularly the Internet, or by means of a data carrier.

12. Method according to claim 10 or 11, characterized by storing the collected configuration information in a data file, in particular a plain ASCII file, and transferring the data file from the proprietary network's site to the network service provider's site.

13. Method according to one or more of claims 10 to 12, characterized by scalable database means, in particular an object-oriented and/or relational database, and a corresponding scalable data model.

14. Method according to one or more of claims 10 to 13, characterized by storing performance history information for the network interconnecting infrastructure.

15. Method according to one or more of claims 10 to 14, characterized by monitoring and/or analyzing the configuration, or configuration changes, respectively, and/or performance ot the network interconnecting infrastructure of the proprietary network, particularly based on rules.

16. Method according to one or more of claims 10 to 15, characterized by scheduling the collection of the configuration information.

17. Method according to one or more of claims 10 to 16, characterized by determining, particularly in advance, which configuration information is to be collected.

18. Method according to one or more of claims 10 to 17, characterized by providing a data collection strategy.

19. Method according to one or more of claims 10 to 18, characterized by backing up the configuration information and restoring at least part of the configuration information in case of failure of at least part of the network interconnecting infrastructure.

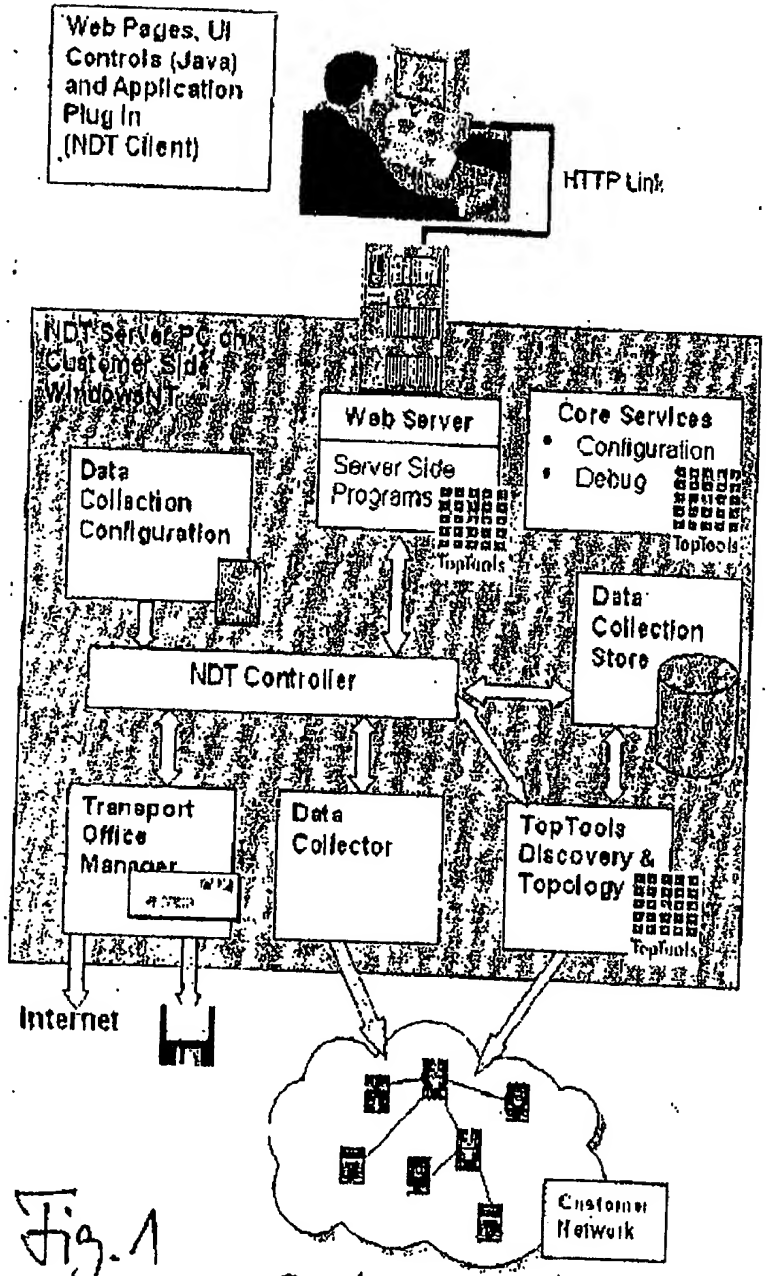
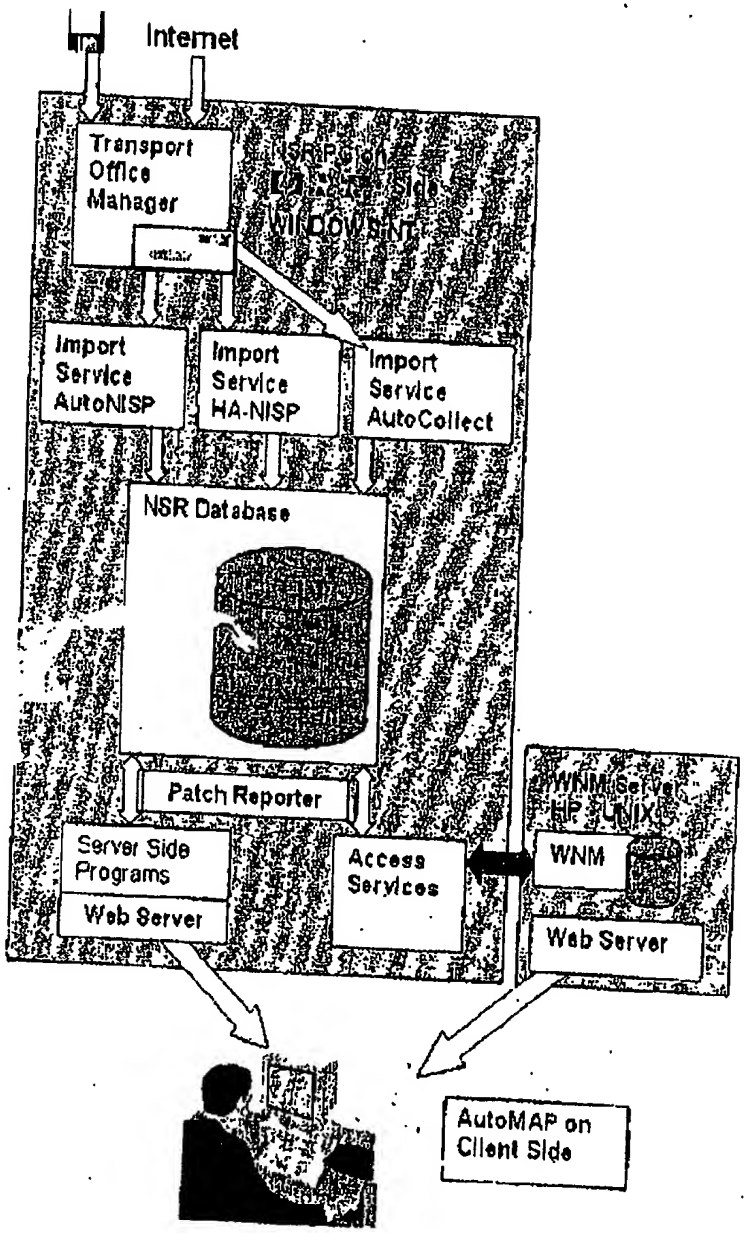


Fig. 1

Customer site



Network Service Provider site

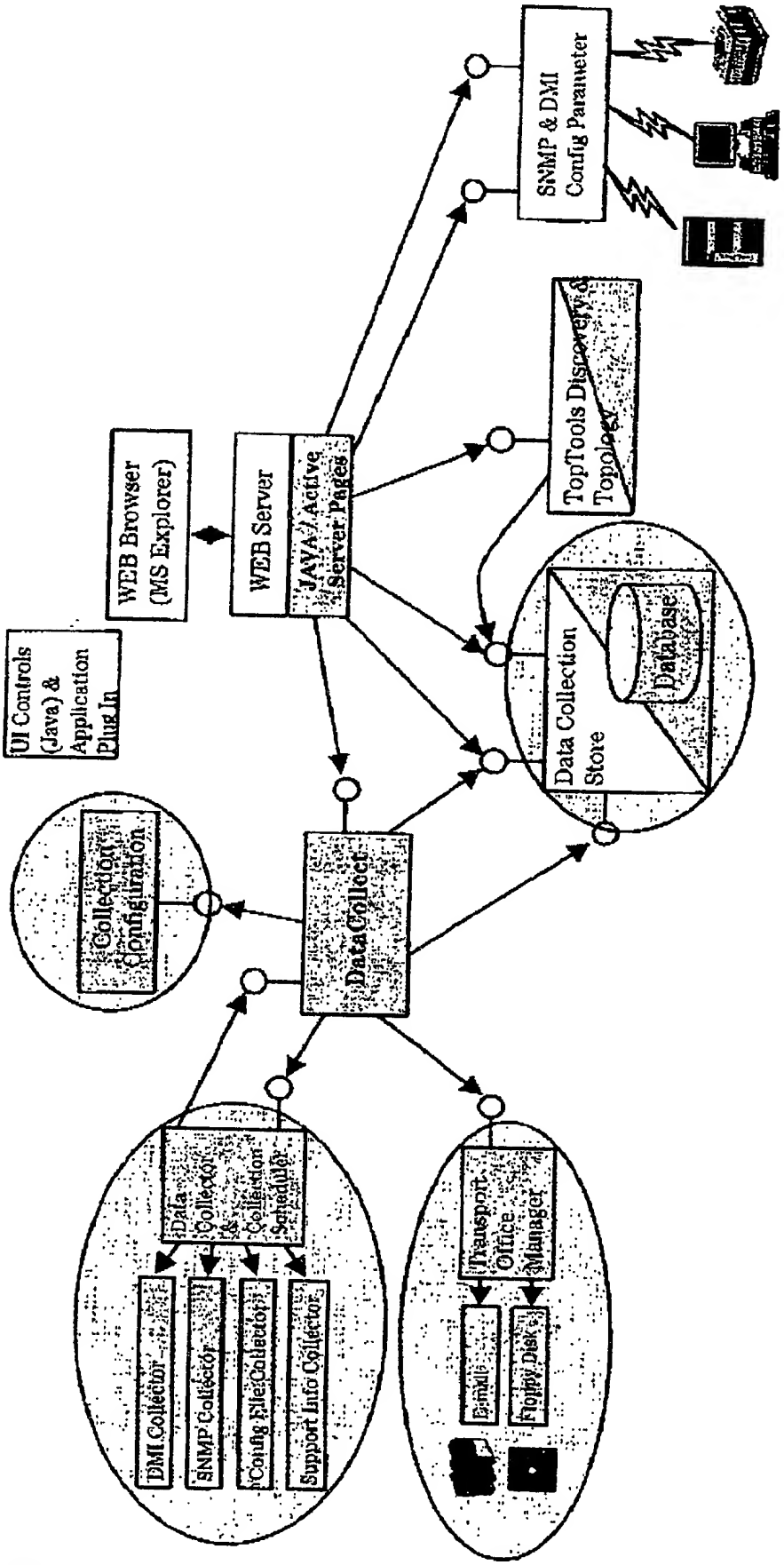

SNMP & DMI Config Parameter
TopTools Discovery & Topology
WEB Browser (MS Explorer)
WEB Server
JAVA / Active Server Pages
UI Controls (Java) & Application Plug In
Data Collection Store
Database
Collection Configuration
DataCollect
Data Collector
Collection Scheduler
DMI Collector
SNMP Collector
Config File Collector
Support Info Collector
Transport
Office Manager
E-mail
Floppy Disk

Fig. 2

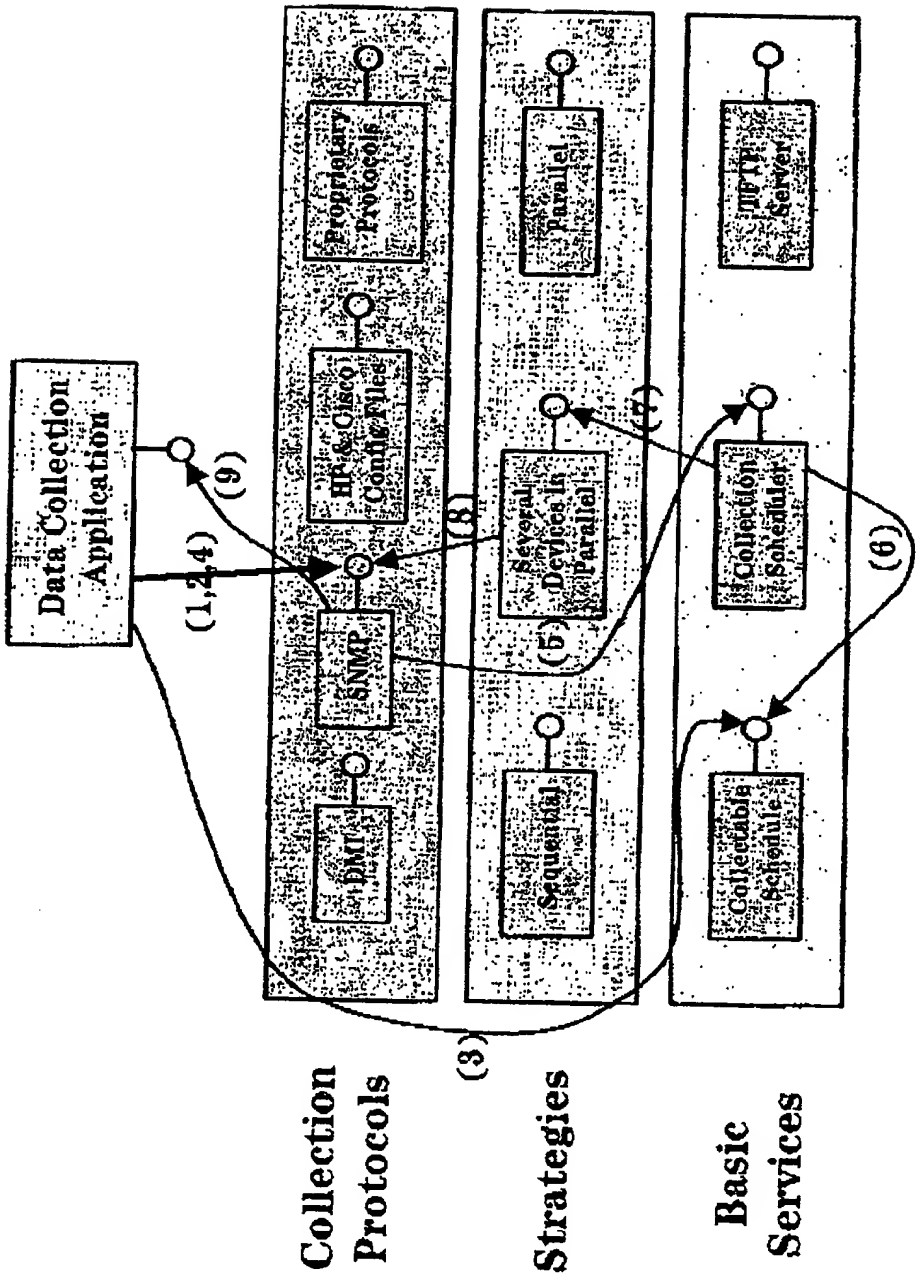

Data Collection Application
(1,2,4)
(9)
(8)
(5)
(7)
(6)
(3)
SNMP
DMI
HP & Cisco Config Files
Proprietary Protocols
Several Devices in Parallel
Parallel
Sequential
Collection Scheduler
Collectable Schedule
HTTP Server
Collection Protocols
Strategies
Basic Services

Fig. 3

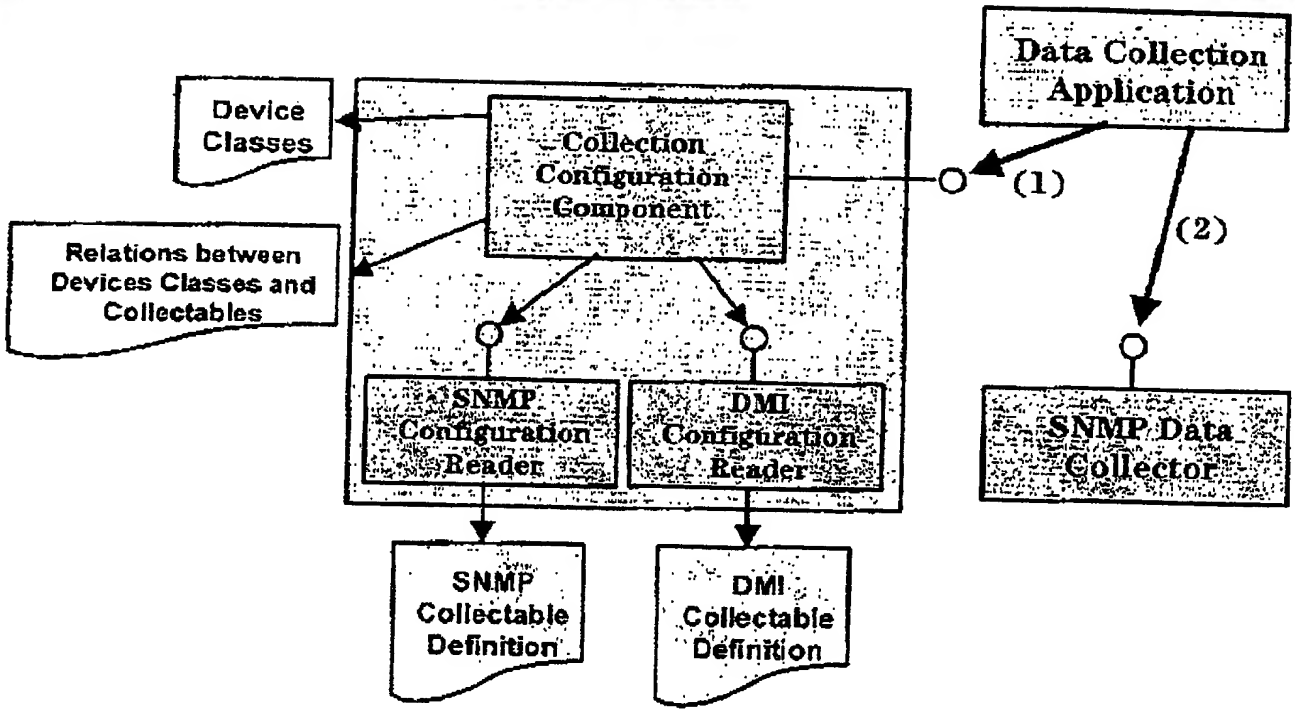

Data Collection Application
Device Classes
Collection Configuration Component
(1)
(2)
Relations between Devices Classes and Collectables
SNMP Configuration Reader
DMI Configuration Reader
SNMP Data Collector
SNMP Collectable Definition
DMI Collectable Definition

Fig. 4

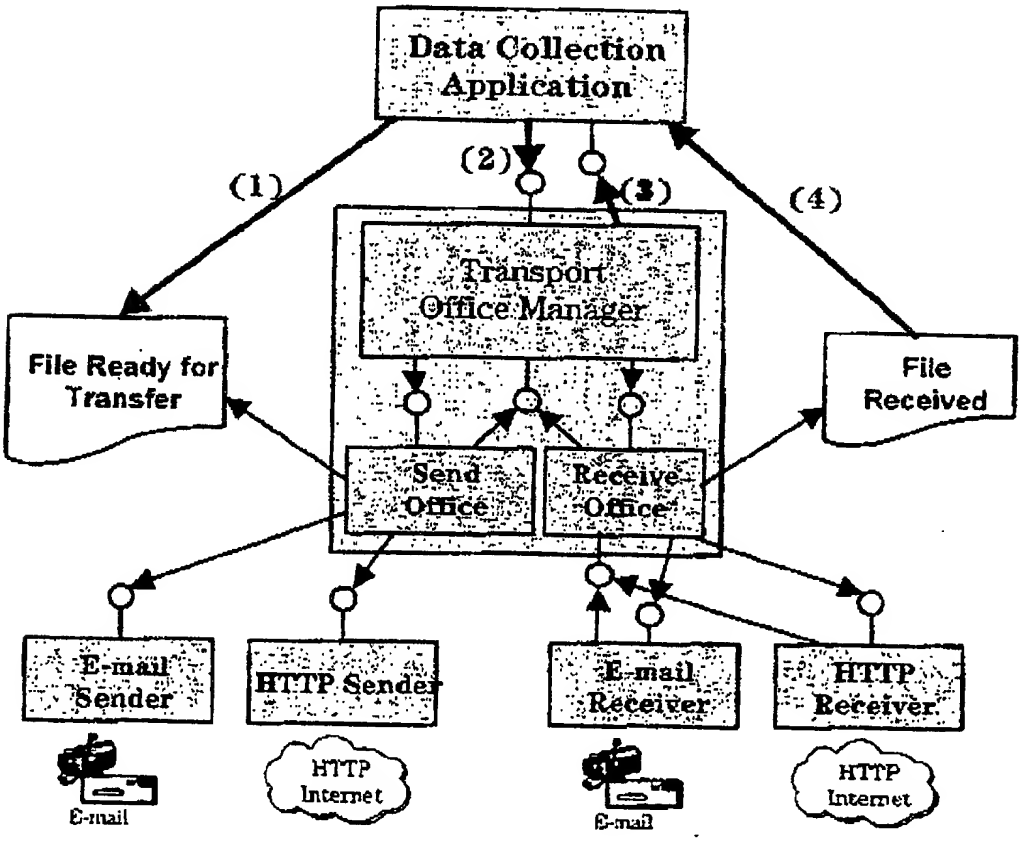

Data Collection Application
(1)
(2)
(3)
(4)
Transport Office Manager
File Ready for Transfer
File Received
Send Office
Receive Office
E-mail Sender
HTTP Sender
E-mail Receiver
HTTP Receiver
E-mail
HTTP Internet
E-mail
HTTP Internet

Fig. 5

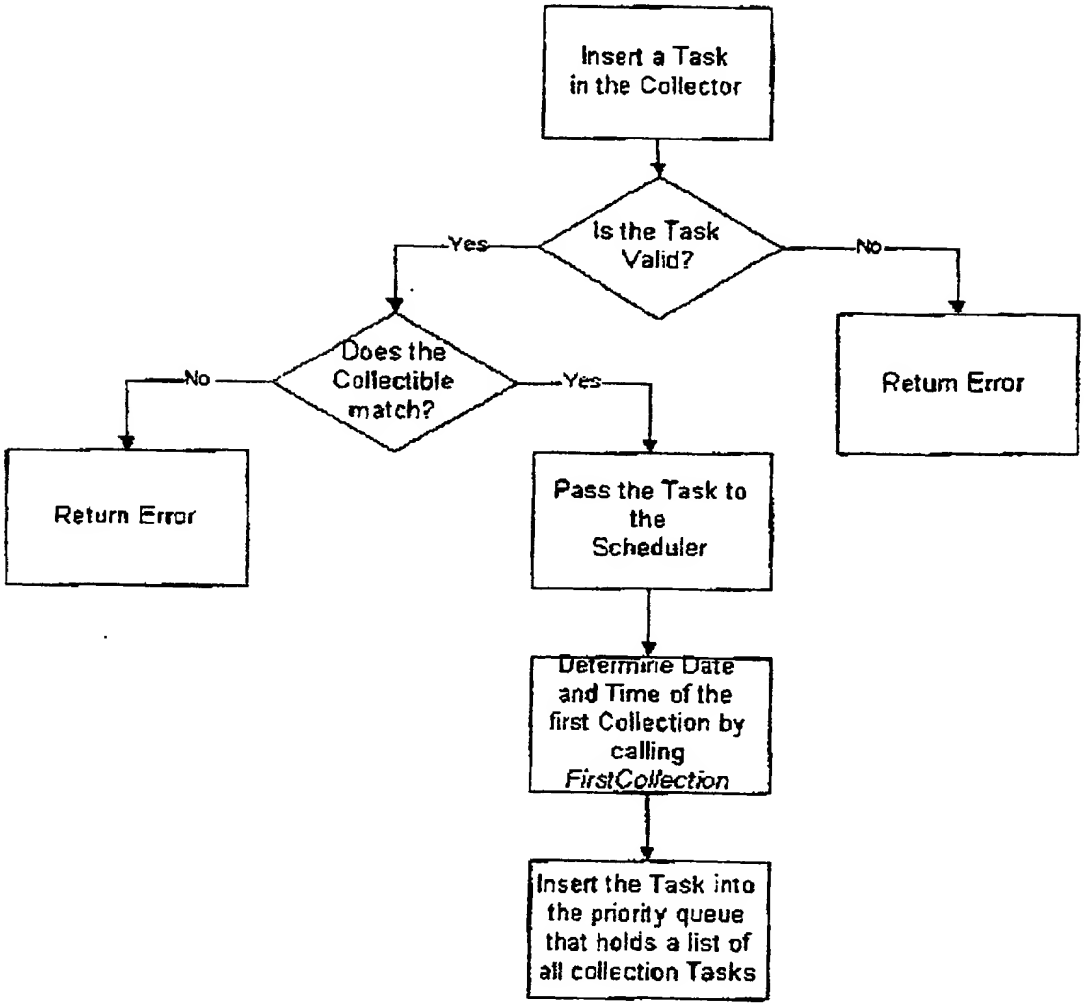

Insert a Task in the Collector
Is the Task Valid?
Yes
No
Return Error
Does the Collectible match?
No
Yes
Return Error
Pass the Task to the Scheduler
Determine Date and Time of the first Collection by calling FirstCollection
Insert the Task into the priority queue that holds a list of all collection Tasks

Fig. 6

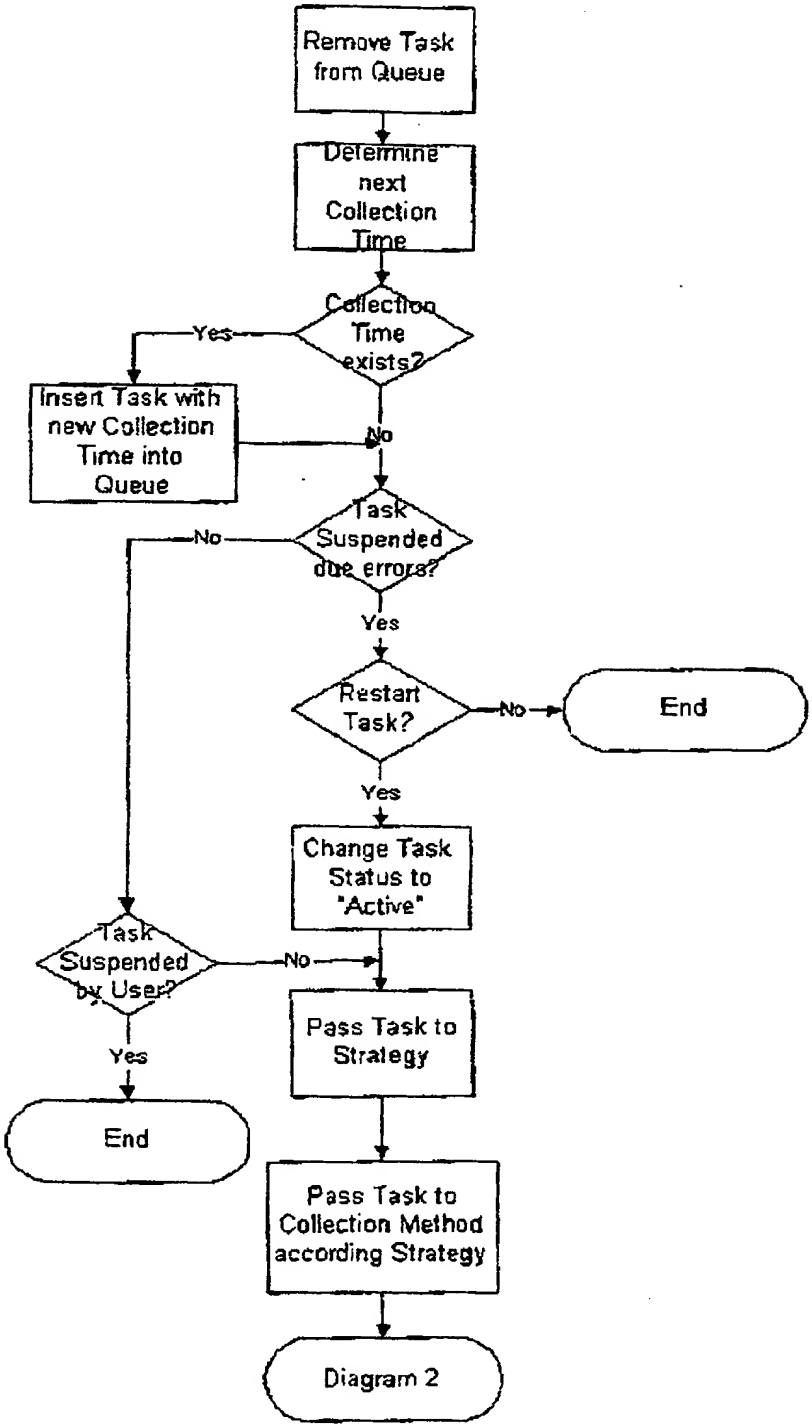

Remove Task from Queue
Determine next Collection Time
Collection Time exists?
Yes
Insert Task with new Collection Time into Queue
No
Task Suspended due errors?
No
Yes
Restart Task?
No
End
Yes
Change Task Status to "Active"
Task Suspended by User?
No
Yes
End
Pass Task to Strategy
Pass Task to Collection Method according Strategy
Diagram 2

Fig. 7a

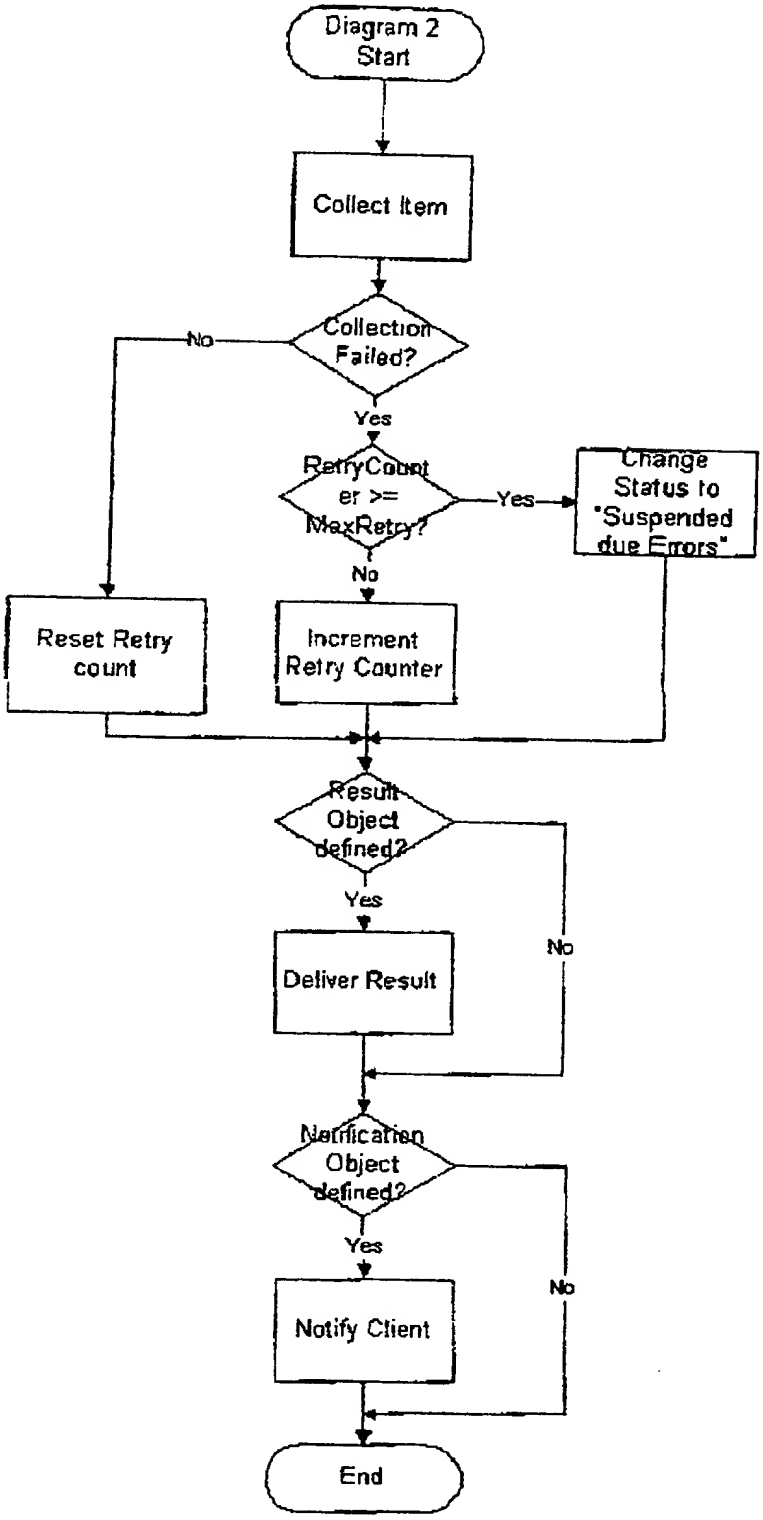


Diagram 2 Start
Collect Item
Collection Failed?
No
Yes
RetryCount er >= MaxRetry?
Yes
Change Status to "Suspended due Errors"
No
Reset Retry count
Increment Retry Counter
Result Object defined?
Yes
No
Deliver Result
Notification Object defined?
Yes
No
Notify Client
End

Fig. 7b

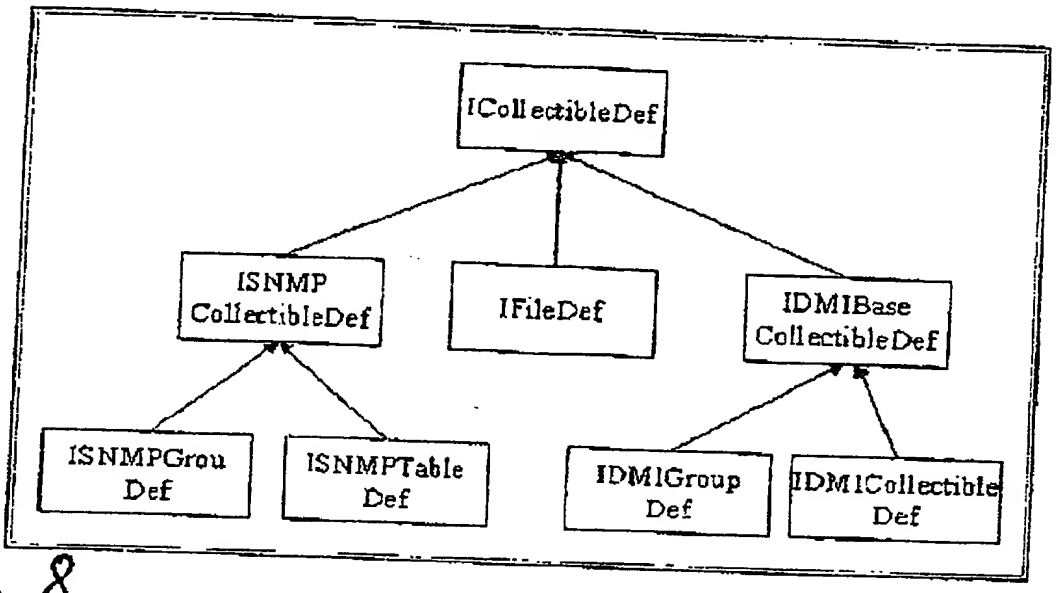


Fig. 8

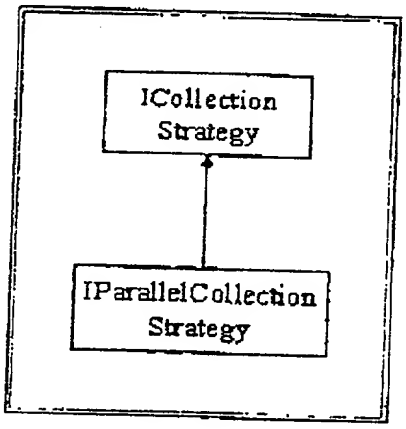


Fig. 9a

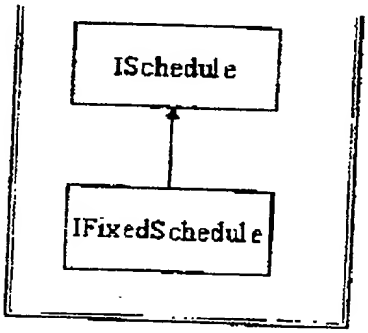


Fig. 9b

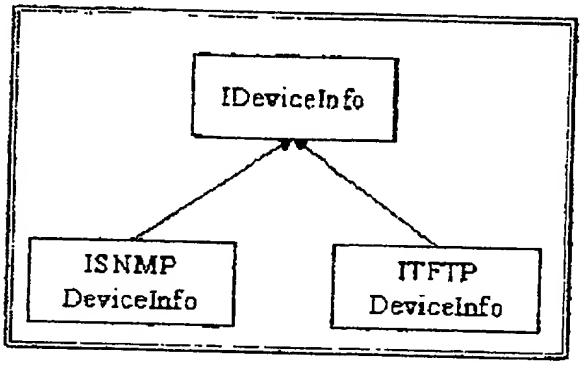


Fig. 9c

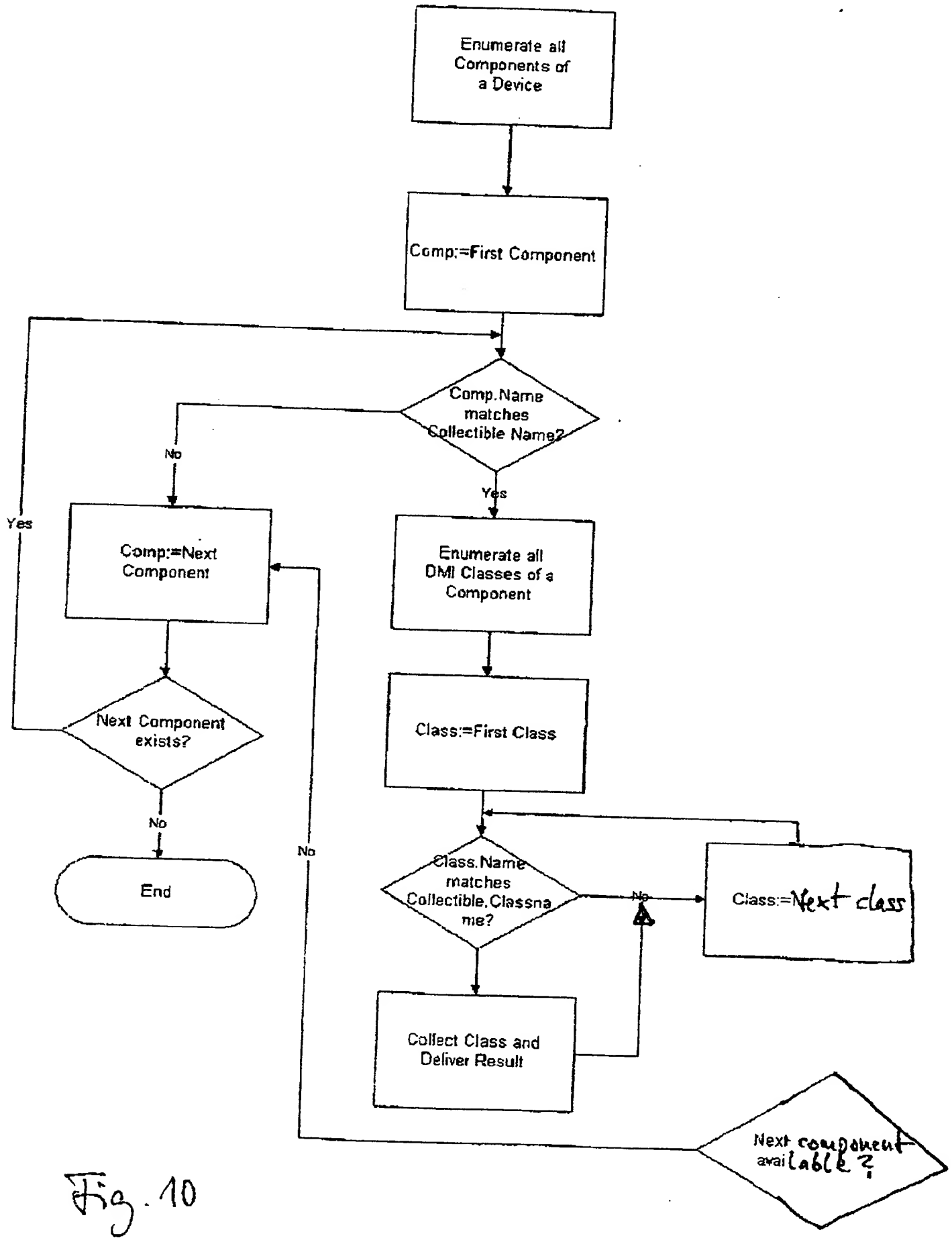

Enumerate all Components of a Device
Comp:=First Component
Comp.Name matches Collectible Name?
No
Yes
Yes
Comp:=Next Component
Enumerate all DMI Classes of a Component
Next Component exists?
Class:=First Class
No
End
No
Class.Name matches Collectible.Classname?
No
Class:=Next class
Collect Class and Deliver Result
Next component available?

Fig. 10